JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

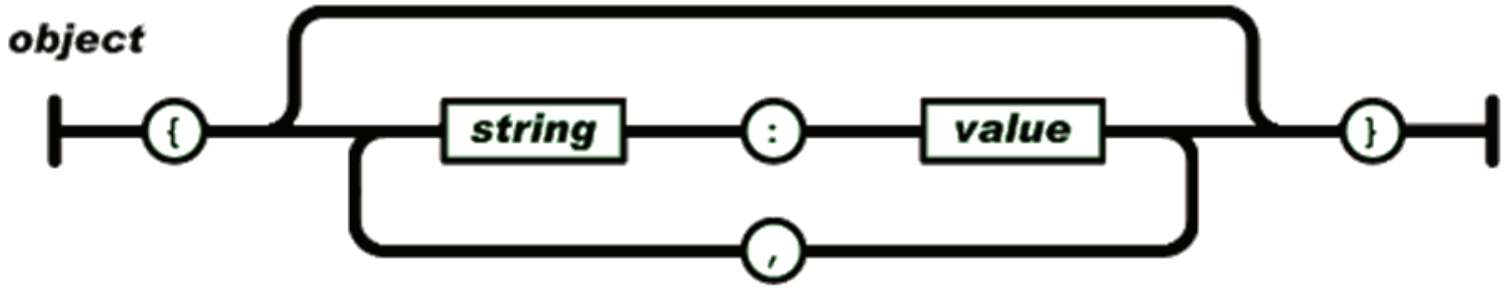
JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).



The following example shows the JSON representation of an object that describes a person.

```
{
     "firstName": "John",
     "lastName": "Smith",
     "age": 25,
     "address": {
         "streetAddress": "21 2nd Street",
         "city": "New York",
         "state": "NY",
         "postalCode": "10021"
     },
     "phoneNumber": [
         { "type": "home", "number": "212 555-1234" },
         { "type": "fax", "number": "646 555-4567" }
     ]
 }
```

A possible equivalent for the above in XML could be:

```
<Person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <age>25</age>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumber type="home">212 555-1234</phoneNumber>
  <phoneNumber type="fax">646 555-4567</phoneNumber>
</Person>
```

JSON in JavaScript

- JSON is a subset of the object literal notation of JavaScript. Since JSON is a subset of JavaScript, it can be used in the language with no muss or fuss.

```
var myJSONObject = {"bindings": [
        {"ircEvent": "PRIVMSG", "method": "newURI", "regex": "^http://.*"},
        {"ircEvent": "PRIVMSG", "method": "deleteURI", "regex": "^delete.*"},
        {"ircEvent": "PRIVMSG", "method": "randomURI", "regex": "^random.*"}
] };
```

- In this example, an object is created containing a single member "bindings", which contains an array containing three objects, each containing "ircEvent", "method", and "regex" members.

- Members can be retrieved using dot or subscript operators.

```
myJSONObject.bindings[0].method // "newURI"
```

```jsp
<%@page contentType="text/html; charset=UTF-8"%>
<%@page import="org.json.simple.JSONObject"%>
<%
   JSONObject obj=new JSONObject();
   obj.put("name","foo");
   obj.put("num",new Integer(100));
   obj.put("balance",new Double(1000.21));
   obj.put("is_vip",new Boolean(true));
   obj.put("nickname",null);
   out.print(obj);
   out.flush();
%>
```

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>

<script type="text/javascript">
function createXMLHttpRequest(){
  // See http://en.wikipedia.org/wiki/XMLHttpRequest
  // Provide the XMLHttpRequest class for IE 5.x-6.x:
  if( typeof XMLHttpRequest == "undefined" ) XMLHttpRequest = function() {
    try { return new ActiveXObject("Msxml2.XMLHTTP.6.0") } catch(e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP.3.0") } catch(e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP") } catch(e) {}
    try { return new ActiveXObject("Microsoft.XMLHTTP") } catch(e) {}
    throw new Error( "This browser does not support XMLHttpRequest." )
  };
  return new XMLHttpRequest();
}

var AJAX = createXMLHttpRequest();

function handler() {
  if(AJAX.readyState == 4 && AJAX.status == 200) {
      var json = eval('(' + AJAX.responseText +')');
      alert('Success. Result: name => ' + json.name + ',' + 'balance => ' + json.balance);
  }else if (AJAX.readyState == 4 && AJAX.status != 200) {
    alert('Something went wrong...');
  }
}

function show(){
  AJAX.onreadystatechange = handler;
  AJAX.open("GET", "service.jsp");
  AJAX.send("");
};
</script>

<body>
  <a href="#" onclick="javascript:show();"> Click here to get JSON data from the server sid
</html>
```